MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2016/2017

TSN2101 – OPERATING SYSTEMS
TOS2111 – OPERATING SYSTEMS I

(All sections / Groups)

19 OCTOBER 2016
9.00 am - 11.00 am
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This Question paper consists of **NINE pages (excluding this page)** with **SEVEN questions.**

2. Attempt any **FIVE** out of **SEVEN** questions. Each question carries **12 marks** and the distribution of the marks for each subdivision is given. Maximum allotted are **60 marks.**

3. Please write all your answers in the Answer Booklet provided.

*Answer only **FIVE** out of **SEVEN** questions. Each question carries 12 marks and the distribution of the marks for each subdivision is given.* ***(5×12=60 marks)***

QUESTION 1:

a. State the steps involved in the '**booting process**' of a computer using general purpose operating system, when a CPU is powered up or reset. **(3 marks)**

b. In an **operating system model handling symmetric multiprocessors**, one copy of the operating system resides in memory, but any CPU can execute that code. State any one **advantage** and one **disadvantage** of using this model. **(1+1=2 marks)**

c. **System calls** provide an interface to the services made available by an operating system.

(i) Briefly explain the steps involved in the **implementation of system calls** for most programming languages. **(2 marks)**

(ii) Three general methods can be used to **pass parameters** to the operating system: (i) Through registers (Register method) (ii) Through an address of the block of parameters that is stored in a table in memory (Block method) (iii) Through pushing parameters onto the stack by the program and popping off the stack by the operating system (Stack method). State the two **limitations** of **Register method** compared to Block or Stack method. **(2 marks)**

d. A **medium-term scheduler** can be used to reduce the degree of multiprogramming or to have a good mix of I/O bound and CPU-bound processes in memory. Answer the following:

(i) What is meant by the '**degree of multiprogramming**'? **(1 mark)**
(ii) What will happen if all processes in memory are **CPU bound**? **(1 mark)**
(iii) What will happen if all processes in memory are **I/O bound**? **(1 mark)**

Continued...

QUESTION 2:

a. Modern operating systems provide features enabling a process to contain multiple threads of control. Answer the following:

(i) State any TWO components that is **shared among all threads** of the same process. **(2 marks)**

(ii) Consider a multithreading model (One-to-One Model) that implements the **thread management in the kernel**. State any one **advantage** and one **disadvantage** of such model. **(2 marks)**

b. Consider the following set of processes, with the length of the CPU-burst time and arrival time given in milliseconds:

Process	Burst-time	Arrival time	Priority
P_0	8	0	3
P_1	6	3	2
P_2	3	4	1
P_3	1	5	3

(i) Draw the **Gantt charts** illustrating the execution of these processes using the following **CPU Scheduling algorithms**:

- **Pre-emptive Shortest Job First** **(2 marks)**
- **Round Robin (quantum = 3)** **(2 marks)**
- **Non Pre-emptive Priority** **(2 marks)**

Assume that smaller numbers represent higher priority and also ignore the priority column for Preemptive Shortest Job First and Round Robin algorithms.

(ii) With respect to **Round Robin** CPU scheduling algorithm, fill up the following table using the above example. **(4×0.5 = 2 marks)**

Process	Waiting Time (ms)
P_0	
P_1	
P_2	
P_3	

Continued...

QUESTION 3:

a. There are three requirements that must be satisfied by any solution for the **critical-section problem**.

(i) State and briefly explain the **THREE requirements**. **(3 marks)**

(ii) A two-process software based solution to the critical-section problem was developed by **Peterson**. The two processes, P_0 and P_1, share the following variables.

```
int turn;
boolean flag [2]; /* initially false*/
```

The structure of process P_i (i == 0 or 1) is shown below. The other process is P_j (j == 1 or 0)

```
do {
        flag[i] = TRUE;
        turn = j;
        while (flag[j] == TRUE && turn == j);
              critical section
        flag[i] = FALSE;
              remainder section
} while (TRUE);
```

Explain how this algorithm satisfies all the three requirements for the critical-section problem with two processes. **(1.5 marks)**

b. Consider the following system with **FOUR** processes and **THREE** resource types:

Resource Type	Number of Instances in the system
R_1	15
R_2	12
R_3	16

Continued...

The snapshot of the system at time T_1 is given below:

Processes	Max.			Allocation			Need		
	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3
P_0				5	2	3	8	5	12
P_1				3	4	6	2	1	3
P_2				1	2	2	1	0	1
P_3				2	3	3	7	6	10

Available		
R_1	R_2	R_3

Answer the following questions using the **banker's algorithm** for **deadlock avoidance**:

(i) In the above snapshot of the system, identify the matrix ***Max.*** (maximum demand of each process at the time of creation) and the vector ***Available*** (number of available resources of each type after allocation)?

(1.5 marks)

(ii) Identify whether the system is **safe or not** by applying the **safety algorithm**. If the system is in a safe state, identify the order in which the processes can complete execution by showing the steps needed.

(2 marks)

(iii) Assume that the following request from process P_0 has arrived at time T_2.

$Request_0$		
R_1	R_2	R_3
3	0	1

Apply **Resource-Request algorithm** to determine, whether this request can be granted immediately or the process P_0 must wait, by showing the steps involved.

(4 marks)

Continued...

QUESTION 4:

a. Briefly explain the difference between **contiguous** and **non-contiguous** memory allocation schemes with **TWO examples** each. **(4 marks)**

b. Consider a memory with a **contiguous allocation scheme** using **best-fit strategy** for satisfying a process request of size n from the list of free holes. Status of memory is as shown below before additional requests from the processes with sizes 10K(P_1), 12K(P_2), 4K(P_3), 6K(P_4) are received in that order. State how would you **allocate memory** to these processes.

Hole1 *(16K)*	Used1 (10K)	*Hole2* *(20K)*	Used2 (5K)	*Hole3* *(5K)*	Used3 (4K)	*Hole4* *(10K)*	Used4 (12K)	*Hole5* *(8K)*

(2 marks)

c. Consider a system using **paging** as a memory-management scheme. Assume a logical memory consists of 32 bytes and physical memory consists of 64 bytes. Page size is given as 4 bytes. Page Table (Page index along with frame number) stored in memory is as given below.

0	3
1	8
2	6
3	2
4	15
5	1
6	12
7	0

Answer the following:

(i) Specify the **number of bits** required for the **logical address,** the **page number**, and the **offset**. **(1.5 marks)**

(ii) Specify the **number of bits** required for the **physical address**, the **frame number**, and the **offset**. **(1.5 marks)**

(ii) Assume the logical addresses generated by CPU are as given below, complete the table by identifying the corresponding physical addresses in binary and decimal. **(3 marks)**

Logical address (Decimal)	**Physical address (Binary)**	**Physical address (Decimal)**
8		
15		

Continued...

QUESTION 5:

a. Consider a **demand-paging system** with the following time requirements:

- Average Access and Transfer time for paging disk: 6000 microseconds
- Average Access time for main memory: 30 microseconds
- Search time with Translation Look-aside Buffer (TLB): 0.01 microseconds

Consider the following scenarios to reference a page containing the instruction/data.

- 91% of the time, the page is found in main memory and page-table entry is found in TLB (TLB hit). This requires one TLB search and one main memory access for each reference.

- 4% of the time, the page and page-table entry are found in main memory (TLB miss). This requires one TLB search and two main memory accesses for each reference.

- 5% of the time, the page fault occurs and the page is found in paging disk.

    - For 60% of that time, assume a free frame is available or the page to be swapped out has not been modified and this requires two TLB searches, two main memory accesses, and one paging disk access for each reference.

    - For 40% of that time, assume the page to be swapped out has been modified and this requires two TLB searches, two main memory accesses, and two paging disk accesses for each reference.

Identify the **effective access time** (in microseconds) needed to satisfy the requests considering the above scenarios. **(4 marks)**

Continued...

b. Assume that the following **address sequence** (first digit represents the page number and the last two digits represent the offset within the page) is recorded in order by tracing a particular process.

201, 392, 345, 674, 600, 699, 035, 302, 456, 469, 210, 345, 307,
463, 407, 303, 325, 312, 435, 236, 666, 604, 360, 240, 230, 360

(i) Considering that we need to look only the page number instead of the entire address and also if we have a reference to a page p, then any references to page p that immediately follow will never cause a page fault, **reduce** the above address sequence to the corresponding **page reference string.** **(1 mark)**

(ii) Using the above identified page reference string for a memory with **FOUR frames** and also by assuming all the frames are initially empty, identify the number of page faults that would occur for **Least Recently Used (LRU) page replacement algorithm**? Show the steps involved and identify the final pages in the frames. **(4 marks)**

c. Assume the total number of available frames is 35 in main memory. How many frames you will allocate for each of the two processes (P_0 with size of 30 pages and P_1 with the size of 120 pages) using **proportional allocation scheme**. **(1 mark)**

d. A process is '**thrashing**', if it is spending more time in paging than execution.

(i) State the **reason** for this problem of 'thrashing'.

(ii) State **a way** that can be used by the system to **eliminate this problem**, once it is detected. **(1+1=2 marks)**

Continued...

QUESTION 6:

a. (i) State any FOUR types of information associated with an **open file**. **(2 marks)**

(ii) State the major **difference** between **direct access method** and **index method** of accessing the information in the file. **(2 marks)**

b. Consider a file currently consisting of **21 blocks** and the system uses **linked strategy** for allocating disk space. Assume the following conditions:

- The location of the first block of the file is already in memory.
- The block information to be added is already stored in memory.
- There is no room to grow at the beginning but there is a room to grow at the end.
- In calculating the number of I/O operations, if the location of the first block of a file changes, do not include the output operations needed to rewrite that revised information to the directory entry on the disk.
- Do not include any I/O operations needed to add a removed block to the free list.
- **The middle block is the 11th block.**

Calculate the **number of read and write operations required** for each of the following and state the reason for your answer: **(4 marks)**

(i) Adding a block after the middle block.
(ii) Adding a block at the end.
(iii) Removing the beginning block.
(iv) Removing the end block.

c. Assume that the **free-space list** is implemented as a **bit map/bit vector**, while implementing file systems. Consider a disk with a total of 8 blocks, where blocks 0, 3, 5, 7 are free and the rest of the blocks are allocated. Identify the **free-space bit map** for this. **(1 mark)**

d. Assume that a file system contains **6 direct pointers** to data blocks, **2 indirect pointers** and **1 double indirect pointer**. Assume also that the size of the data blocks is 4KB and each block numbers are represented as 8 byte unsigned integers.

(i) Identify the maximum number of bytes addressed by 6 direct pointers? **(1 mark)**

(ii) Identify the maximum number of bytes addressed by 2 indirect pointers? **(1 mark)**

(iii) Identify the maximum number of bytes addressed by 1 double indirect pointer? **(1 mark)**

Continued...

QUESTION 7:

a. A variety of disk-organization techniques called **Redundant Arrays of Independent Disks (RAID)** are commonly used to address performance and reliability issues. Briefly explain the techniques of **striping** and **mirroring** related to this. **(2 marks)**

b. Suppose that a magnetic disk drive has 50 cylinders, numbered from 0 to 49. Assume that the drive is currently serving a request at cylinder 33, and the head is moving towards cylinder 0. The queue of pending requests, in First In First Out order, is

23, 45, 25, 15, 36, 5, 47

Starting from the current head position, what is the **total seek distance (in cylinders)** that the disk arm moves and the **total seek time (in milliseconds)** required to satisfy all the pending requests, for each of the following disk-scheduling algorithms?

(i) Shortest Seek Time First (SSTF) Disk Scheduling Algorithm
(ii) Circular Look Disk Scheduling Algorithm

Assume the seek time required for traversing each cylinder is 300 microseconds.
(3+3=6 marks)

c. Briefly explain how the processor time is allocated to different processes in the **'Completely Fair CPU Scheduler'** provided by **Linux Kernel**. **(2 marks)**

d. **Windows** implements **virtual memory using demand paging with clustering**. Briefly explain how the page faults are handled using this technique. **(2 marks)**

END OF EXAM